

```

//define rochcor.h in your analysis header file, for example, in Anal.h
#include <rochcor2012wasym.h>
#include <muresolution.h>

//=====

//In your analysis code (Anal.C),

void Anal::main()
{

if (fChain == 0) return;
Double_t nentries = fChain->GetEntries();

    rochcor2012 *rmcor = new rochcor2012(); // make the pointer of rochcor class

//for-loop of the event
for (Int_t k=0; k<nentries; ++k){
    fChain->GetEntry(k);

    ....

    TLorentzVector mu; //TLorentzVector of the reconstructed muon-

    //Set TLorentzVector of muon object
    mu.SetPtEtaPhiM(...);

    float qter = 1.0; // added it by Higgs group's request to propagate the uncertainty

    //If you run MC, apply the muon momentum correction, "momcor_mc()" function (only for MC)
    //ntrk (number of track layer) is one of input and it can slightly improved the extra smearing
    rmcor->momcor_mc(mu, charge, ntrk=0, qter);

    //If you run data, apply the muon momentum correction, "momcor_data()" function (only for Data)
    // No run dependence for 2012 data, so default of "runopt=0"
    rmcor->momcor_data(mu, charge, runopt=0, qter);

}
}

//In Root, compile "muresolution.cc" and "rochcor2012wasym.C" before compiling your analysis code
root -l
.L muresolution.cc++ // compile "muresolution.cc"
.L rochcor2012wasym.cc++ // compile "rochcor2012wasym.C"
.L Anal.C+ // compile your analysis code, "Anal.C"
Anal pf // running the main function
pf.main()

```